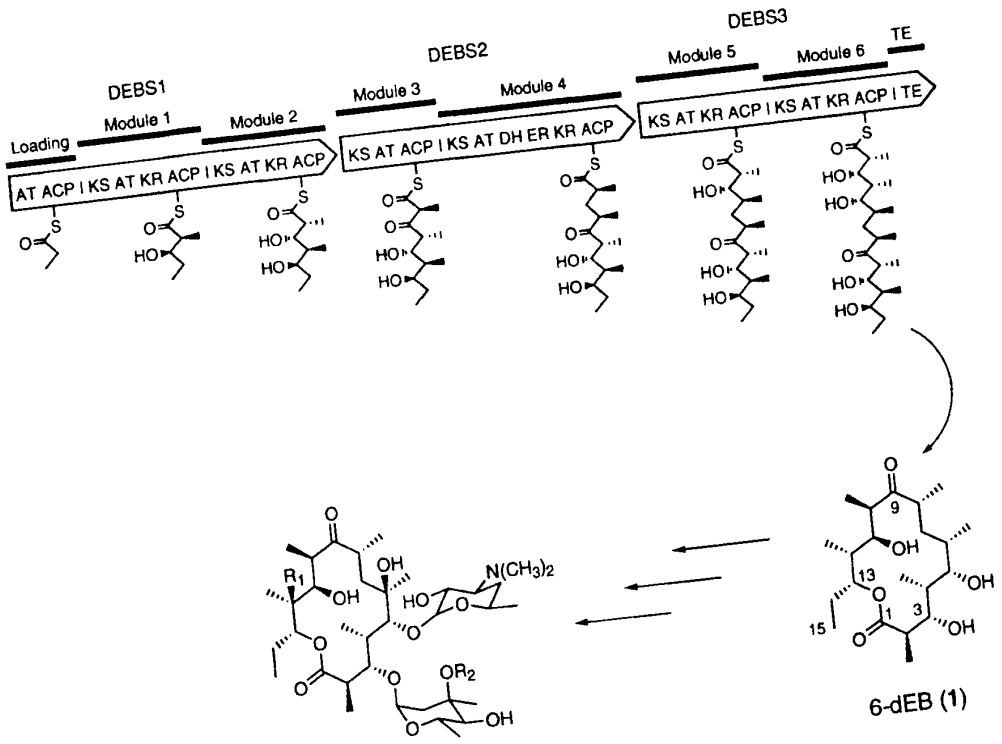
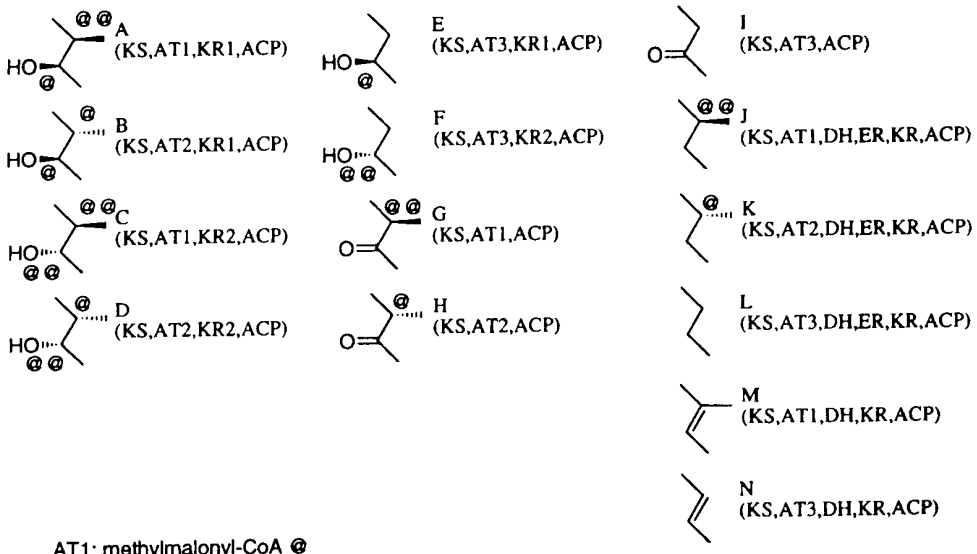

DEBS1
DEBS2
DEBS3
Loading
Module 1
Module 2
Module 3
Module 4
Module 5
Module 6
TE
AT ACP | KS AT KR ACP | KS AT KR ACP
KS AT ACP | KS AT DH ER KR ACP
KS AT KR ACP | KS AT KR ACP | TE
6-dEB (1)
R1=OH, R2=CH3 Erythromycin A (2)
R1=H, R2=CH3 Erythromycin B
R1=R2=H Erythromycin D

Figure 1

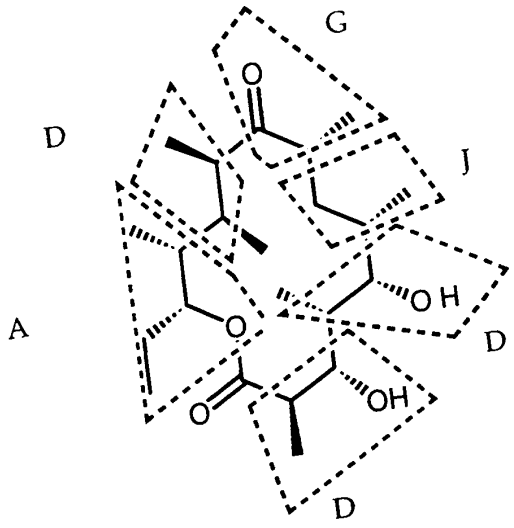


AT1: methylmalonyl-CoA @
AT2: methylmalonyl-CoA @@
AT3: malonyl-CoA

KR1: hydroxyl @
KR2: hydroxyl @@

@@ = counter clockwise
@ = clockwise

Figure 2



CHUCKLES: ADGJDD

SMILES:
C1(=O)-[C@H](C)[C@@H](OH)-[C@@H](C)[C@@H](OH)-
[C@@H](C)C-[C@@H](C)C(=O)-[C@H](C)[C@@H](C)-
[C@@H](C)[C@@H](CC)O1

Figure 3

Figure 4A

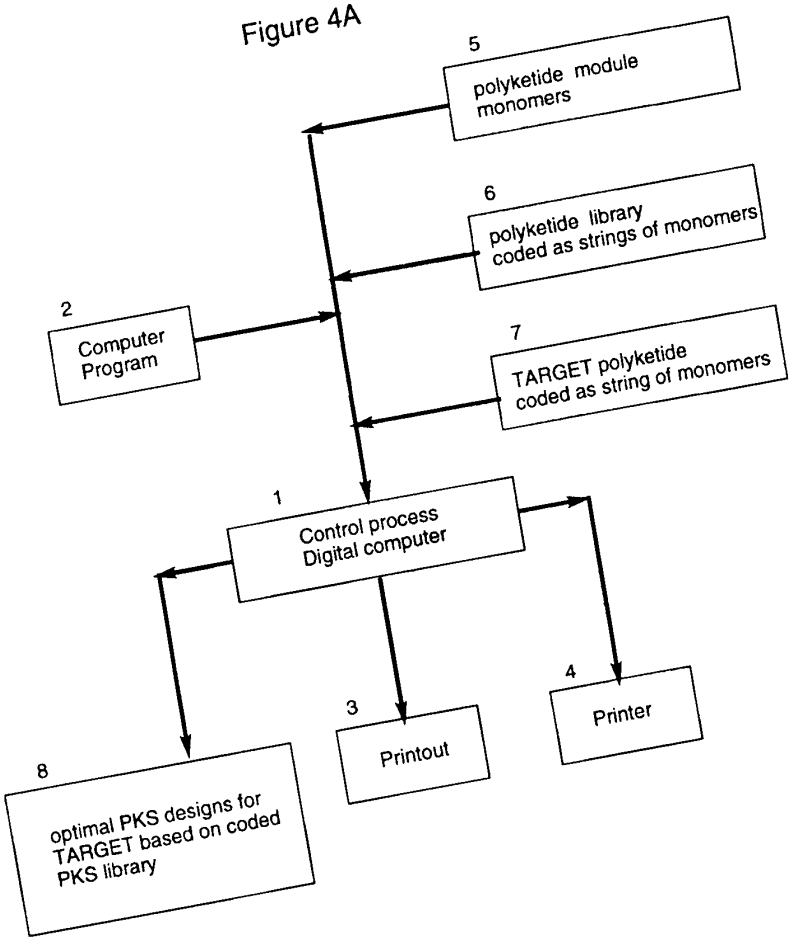

5
polyketide module monomers
6
polyketide library coded as strings of monomers
2
Computer Program
7
TARGET polyketide coded as string of monomers
1
Control process Digital computer
3
Printout
4
Printer
8
optimal PKS designs for TARGET based on coded PKS library

Figure 4B

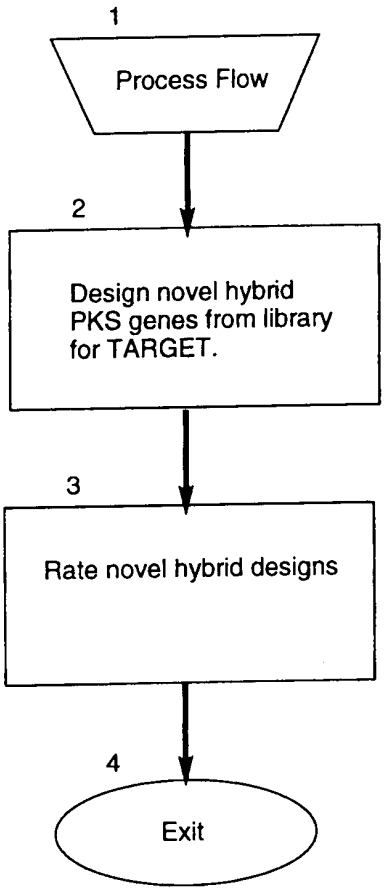
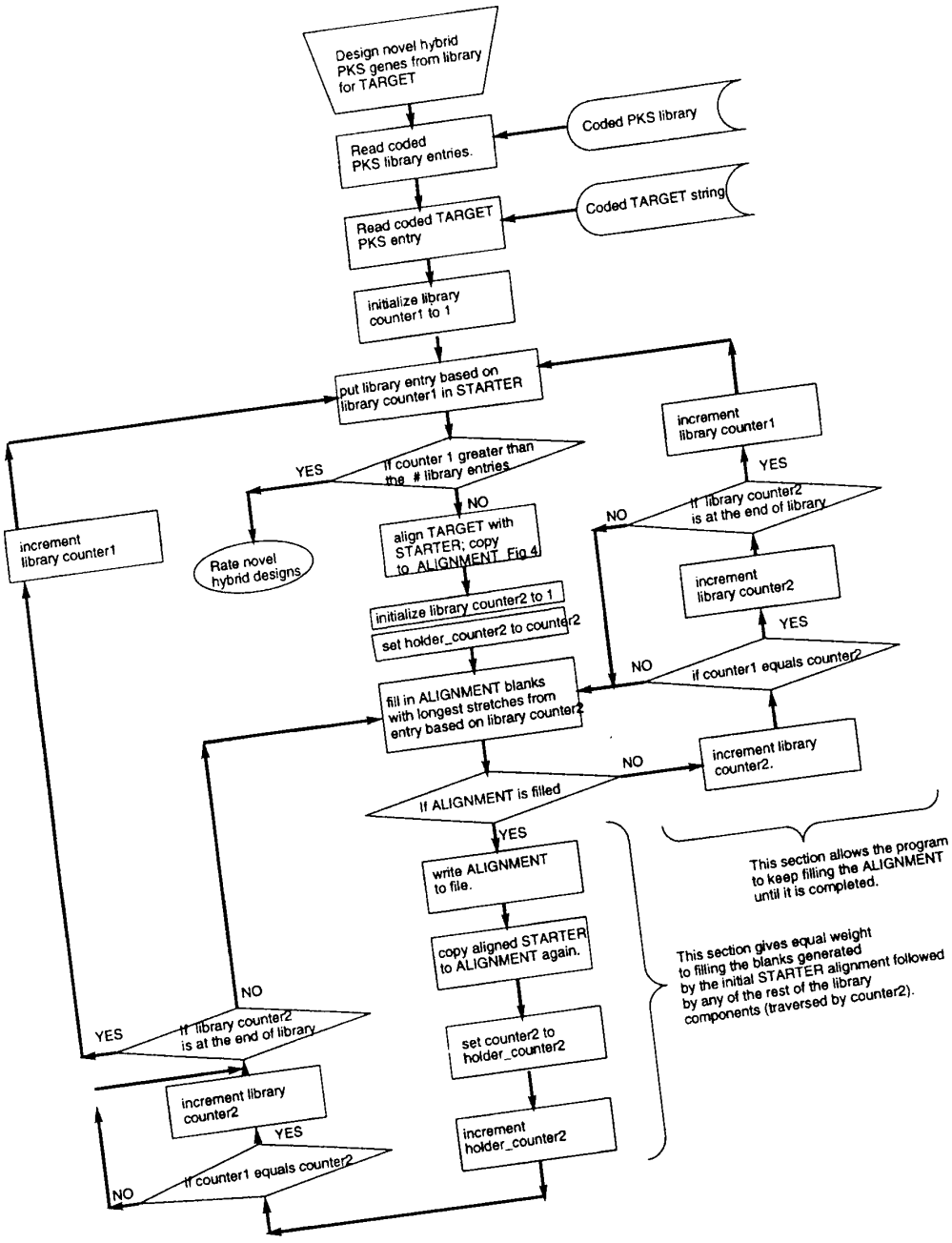

1
Process Flow
2
Design novel hybrid PKS genes from library for TARGET.
3
Rate novel hybrid designs
4
Exit

Figure 4C


Design novel hybrid PKS genes from library for TARGET
Coded PKS library
Read coded PKS library entries.
Coded TARGET string
Read coded TARGET PKS entry
initialize library counter1 to 1
put library entry based on library counter1 in STARTER
If counter 1 greater than the # library entries
YES
NO
increment library counter1
Rate novel hybrid designs
align TARGET with STARTER; copy to ALIGNMENT Fig 4
initialize library counter2 to 1
set holder_counter2 to counter2
fill in ALIGNMENT blanks with longest stretches from entry based on library counter2
If ALIGNMENT is filled
NO
YES
write ALIGNMENT to file.
copy aligned STARTER to ALIGNMENT again.
set counter2 to holder_counter2
increment holder_counter2
If library counter2 is at the end of library
NO
YES
increment library counter2
If counter1 equals counter2
YES
NO
increment library counter2.
If library counter2 is at the end of library
NO
YES
increment library counter1
This section allows the program to keep filling the ALIGNMENT until it is completed.
This section gives equal weight to filling the blanks generated by the initial STARTER alignment followed by any of the rest of the library components (traversed by counter2).

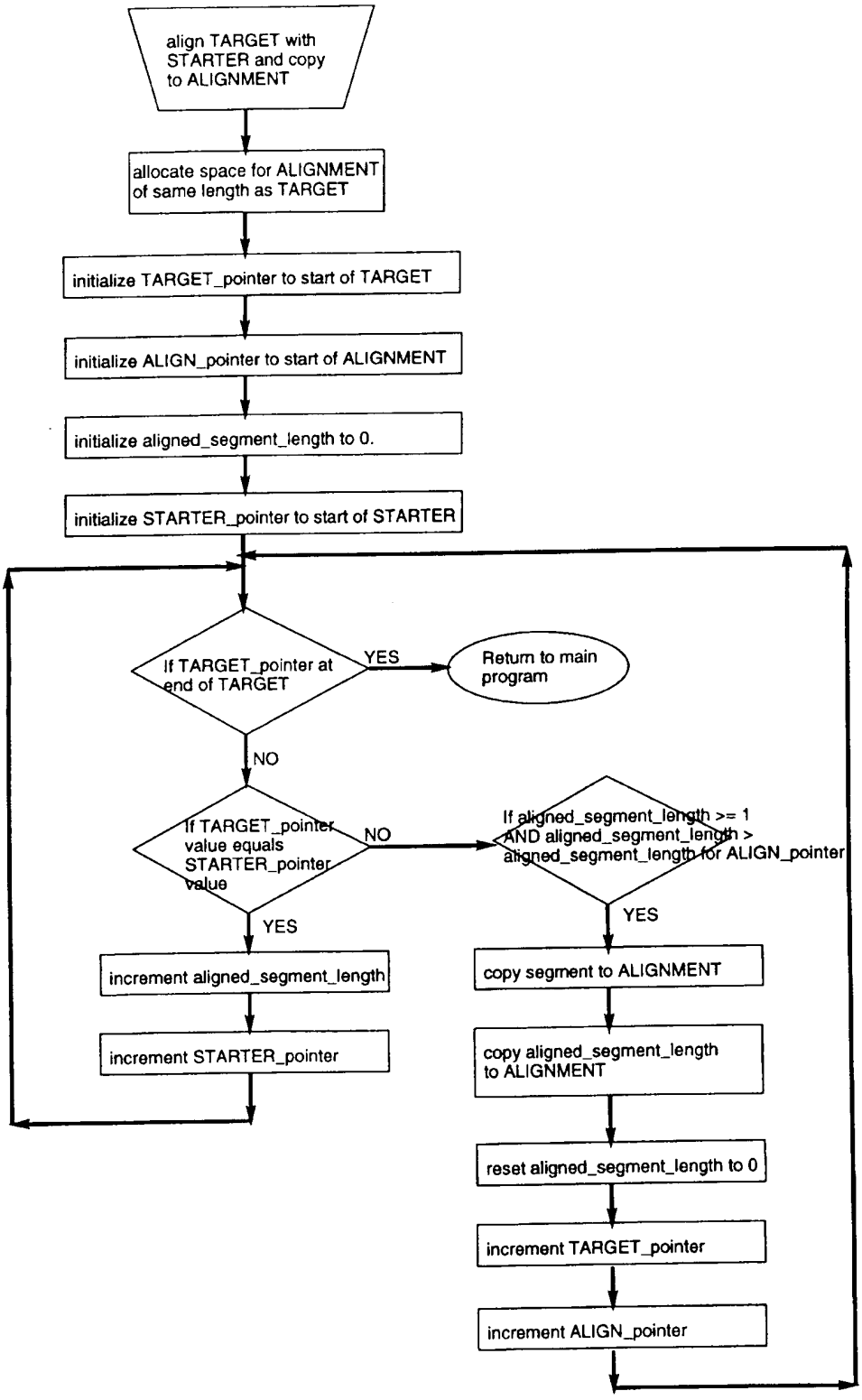

align TARGET with STARTER and copy to ALIGNMENT
allocate space for ALIGNMENT of same length as TARGET
initialize TARGET_pointer to start of TARGET
initialize ALIGN_pointer to start of ALIGNMENT
initialize aligned_segment_length to 0.
initialize STARTER_pointer to start of STARTER
If TARGET_pointer at end of TARGET
YES
Return to main program
NO
If TARGET_pointer value equals STARTER_pointer value
NO
If aligned_segment_length >= 1 AND aligned_segment_length > aligned_segment_length for ALIGN_pointer
YES
YES
increment aligned_segment_length
increment STARTER_pointer
copy segment to ALIGNMENT
copy aligned_segment_length to ALIGNMENT
reset aligned_segment_length to 0
increment TARGET_pointer
increment ALIGN_pointer

Figure 4D

Figure 4E

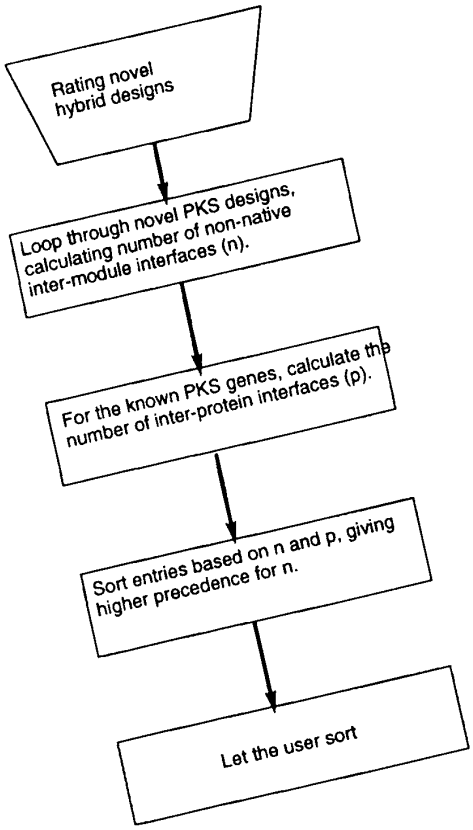
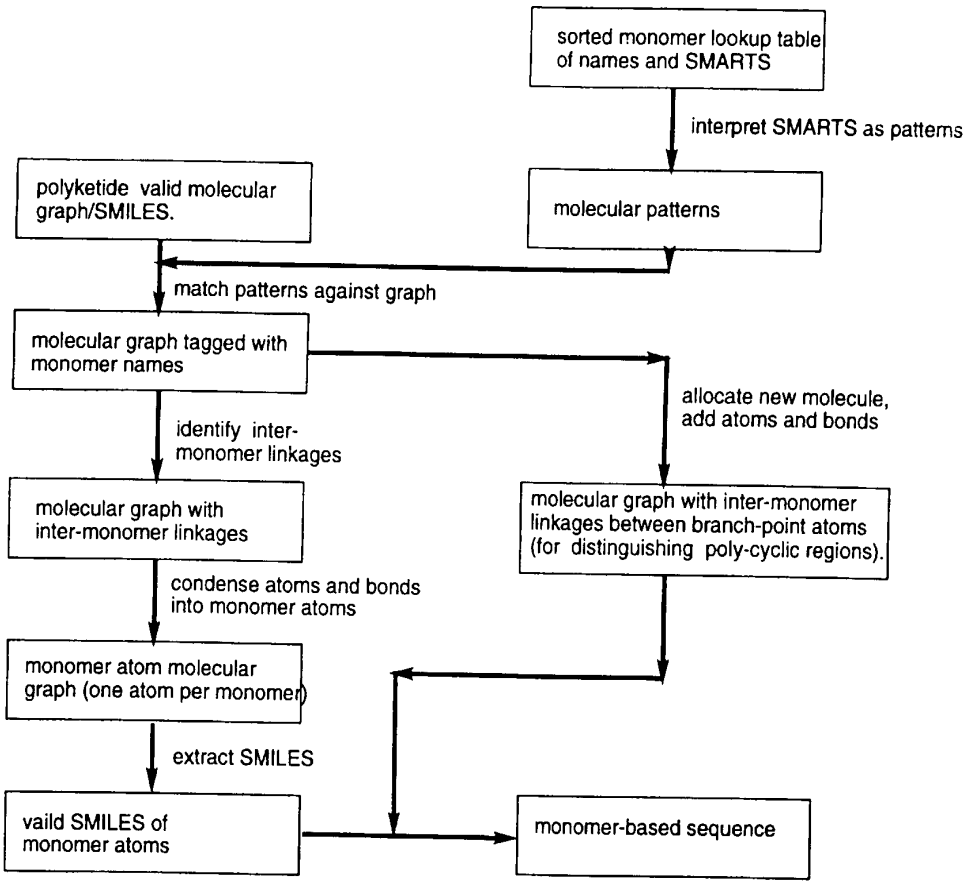

Rating novel hybrid designs
Loop through novel PKS designs, calculating number of non-native inter-module interfaces (n).
For the known PKS genes, calculate the number of inter-protein interfaces (p).
Sort entries based on n and p, giving higher precedence for n.
Let the user sort

Figure 5


sorted monomer lookup table of names and SMARTS
interpret SMARTS as patterns
polyketide valid molecular graph/SMILES.
molecular patterns
match patterns against graph
molecular graph tagged with monomer names
allocate new molecule, add atoms and bonds
identify inter-monomer linkages
molecular graph with inter-monomer linkages
molecular graph with inter-monomer linkages between branch-point atoms (for distinguishing poly-cyclic regions).
condense atoms and bonds into monomer atoms
monomer atom molecular graph (one atom per monomer)
extract SMILES
vaild SMILES of monomer atoms
monomer-based sequence